

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Pavlica

**Implementacija Dinamičnega izenačevalnika v okolju
SuperCollider**

DIPLOMSKO DELO

UNIVERZITETNI INTERDISCIPLINARNI ŠTUDIJSKI
PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN
MATEMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Blaž Pavlica

**Implementacija Dinamičnega izenačevalnika v okolju
SuperCollider**

DIPLOMSKO DELO

UNIVERZITETNI INTERDISCIPLINARNI ŠTUDIJSKI
PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN
MATEMATIKA

MENTOR: doc. dr. Matija Marolt

Ljubljana, 2014

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V diplomskem delu v okolju SuperCollider implementirajte dinamični izenačevalnik zvočnih signalov. Omogoča naj razbitje signala na več frekvenčnih pasov brez popačenja faze in spreminjanje dinamičnega razpona posameznih frekvenčnih pasov. Parametre izenačevalnika naj bo mogoče spreminjati preko ustreznega grafičnega vmesnika.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Blaž Pavlica, z vpisno številko **63100303**, sem avtor diplomskega dela z naslovom:

Implementacija dinamičnega izenačevalnika v programskem okolju SuperCollider

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Matija Marolta,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 23. septembra 2014

Podpis avtorja:

Zahvaljujem se mentorju Matiju Maroltu za pomoč pri izdelavi diplomskega dela, Robertu Rozmanu za predavanje predmeta Digitalno procesiranje signalov in nasvete glede snovanja filtrov ter družini in prijateljem, ki so me podpirali med pisanjem.

Kazalo

Povzetek

Abstract

Poglavje 1	Uvod	1
Poglavje 2	Osnove in orodja.....	3
2.1	Filtri	3
2.2	Izenačevalnik signala	5
2.3	Dinamični razpon signala in njegov procesor.....	6
2.3.1	Dinamični razpon	6
2.3.2	Procesor dinamičnega razpona	6
2.4	KEO filtri, linearna faza.....	8
2.5	Algoritem Parks-McClellan	10
2.6	Večpasovni procesor dinamičnega razpona.....	10
2.7	Matlab	11
2.8	SuperCollider 3.6	11
Poglavje 3	Izvedba.....	13
3.1	Cilj diplomskega dela	13
3.2	Zasnova filtrov	13
3.3	Implementacija dinamičnega izenačevalnika	15
3.4	Grafični vmesnik.....	16
Poglavje 4	Evaluacija dela.....	19
Poglavje 5	Zaključek.....	21

Seznam uporabljenih kratic

kratica	angleško	slovensko
KEO	finite impulse response	končen enotin odziv
NEO	infinite impulse response	neskončen enotin odziv
VPDR	multiband dynamic range processor	Večpasovni procesor dinamičnega razpona

Povzetek

Cilj je narediti dinamični izenačevalnik, torej izenačevalnik, ki ne le statično spreminja frekvenčnega odziva signala, ampak se dinamični razpon posameznih pasov frekvenčnega spektra spreminja glede na razpon le-teh. Povedano drugače, frekvenčni spekter je s filtri razbit na 30 pasov, na vsakem pasu je procesor dinamičnega razpona signala (kompresor, expander, gate). Nastavitve procesorjev so preko grafičnega vmesnika povezane tako, da delovanje in uporaba dinamičnega izenačevalnika spominjata bolj na navaden izenačevalnik kot pa večpasovni procesor dinamičnega signala. Filtri so zasnovani v Matlabu, in sicer tako, da imajo linearno fazo. Enotini odzivi posameznih filtrov so zamaknjeni tako, da imajo sredinski koeficient poravnan, s tem imajo tudi vsi enak fazni zamik, zaradi česar je fazni odziv njihovega seštevka prav tako linearen. Izenačevalnik in grafični vmesnik sta sprogramirana v programskem jeziku SuperCollider 3.6.

Ključne besede: dinamični izenačevalnik, dinamični razpon signala, večpasovni procesor dinamičnega razpona signala, Matlab, linearna faza, grafični vmesnik, SuperCollider 3.6

Abstract

The goal is to build a dynamic equalizer, that is an equalizer that not only statically changes the frequency response of a signal, but the dynamic range of individual bands of the frequency spectrum is changed according to the dynamic range of themselves. In other words, the frequency spectrum is broken up with filters into 30 bands, on each there is a dynamic range processor (compressor, expander, gate). The settings of the processors are connected through a graphic user interface in such a way that its functioning and use resembles more an ordinary equalizer than a multiband dynamic range processor. Filters are designed in Matlab in such a way that they have a linear phase. Impulse responses of individual filters are shifted so that their middle coefficients are aligned. That way they all have the same phase delay, because of which the phase response of the sum of all the filters also has a linear phase. The equalizer and graphic user interface were programmed in SuperCollider 3.6.

Keywords: dynamic equalizer, dynamic range of a signal, multiband dynamic range processor, Matlab, linear phase, graphic user interface, SuperCollider 3.6

Poglavje 1 Uvod

Naloga dinamičnega izenačevalnika je spreminjati dinamični razpon signala le na nekem ali nekaterih delih frekvenčnega spektra. Z navadnim izenačevalnikom spremenimo jakost nekega dela frekvenčnega spektra, to je zadovoljivo v večini primerov. S procesorjem dinamičnega razpona pa spreminjamo dinamični razpon signala glede na sam signal, torej včasih ostane enak, včasih ga stisne, razširi itd. Recimo, da imamo signal, v katerem je le ob nekaterih trenutkih in le na določenem delu spektra preveč signala. Če bi uporabili navaden izenačevalnik, bi lahko zmanjšali jakost signala na tistem delu spektra, a to bi pomenilo, da smo ga zmanjšali tudi, ko ga ni preveč, to pa je lahko nezaželeno. Če bi uporabili procesor dinamičnega razpona, bi zaradi presežka signala v tistem delu spektra procesor stisnil celoten signal, kar je tudi večinoma nezaželeno. Zaradi tega je bil razvit večpasovni procesor dinamičnega razpona, ta frekvenčni razpon signala razbije na nekaj pasov in procesira vsakega posebej. Dinamični izenačevalnik je razširitev tega koncepta, frekvenčni razpon signala razbijemo na še več posameznih pasov in nastavitve procesorjev povežemo tako, da jih lahko nastavljamo s funkcijami, odvisnimi od frekvence. Te funkcije so v grafičnem vmesniku predstavljene kot krivulje, zaradi česar dinamični izenačevalnik bolj spominja na izenačevalnik signala kot pa na večpasovni procesor dinamičnega razpona signala.

Filtre za razbitje signala na posamezne pasove sem zasnoval s programskim okoljem Matlab, in sicer s funkcijo za izvajanje algoritma Parks-McClellan za snovanje filtrov. To je algoritem, ki najde optimalni Chebyshev filter s končnim enotnim odzivom. Metodi podamo želene karakteristike filtra, in sicer: prepustne predele, zaporne predele, dovoljeno deviacijo v posameznih predelih ter red filtra. Če red filtra ne bo dovolj velik, filter ne bo optimalen, zato obstaja tudi funkcija, ki oceni red filtra. Filtri, ustvarjeni s to metodo, imajo linearno fazo, kar je zelo zaželeno zato, da ne prihaja do zvočnih artefaktov, kjer se filtri prekrivajo. Enotni odzivi filtrov so zamaknjeni tako, da so sredinski koeficienti poravnani, s tem imajo vsi enako fazno zakasnitev. To je pomembno, ko so seštet skupaj, da ima končni signal linearno fazo. Koeficiente filtrov sem nato presnel v programsko okolje SuperCollider 3.6. Ta programski jezik je bil sprva zasnovan za realnočasovno sintezo zvoka in algoritmično kompozicijo, a vsebuje dobro okolje za realnočasovno digitalno procesiranje signalov. V SC sem najprej sprogramiral proceduro, ki razdeli signal na več paralelnih signalov in vsakega posebej konvolira s koeficienti zasnovanih filtrov. Na vsakem od paralelnih signalov je procesor

dinamičnega razpona signala, katerega nastavitve lahko nastavljamo z drugimi procedurami. Druga procedura ustvari grafični vmesnik, s katerim upravljamo funkcije, ki določijo nastavitve procesorjev.

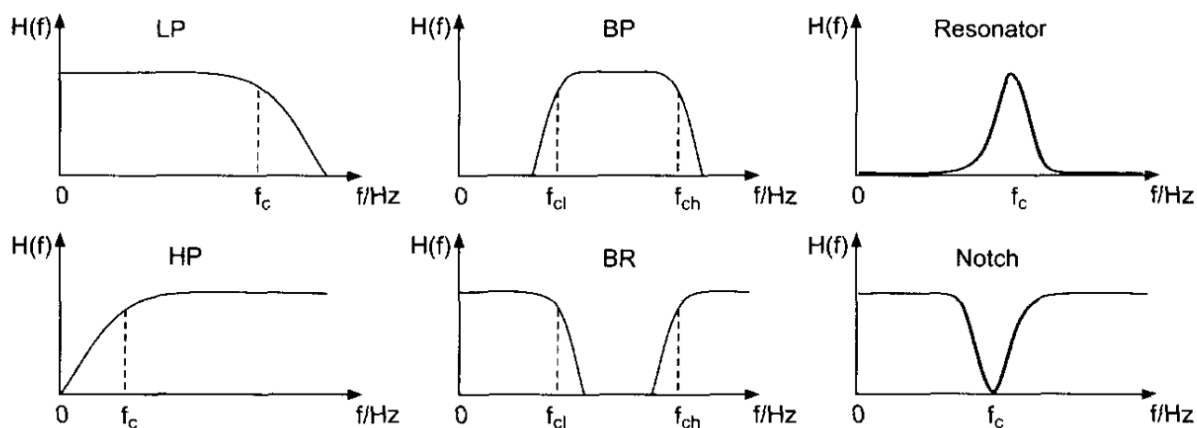
Poglavje 2 Osnove in orodja

2.1 Filtri

Beseda filter ima lahko več različnih pomenov, širši pomen je iz neke množice izluščiti elemente z nekimi zaželenimi lastnostmi. Tu bom opisoval le filtre v domeni procesiranja signalov. Signal lahko opazujemo v časovni ali frekvenčni domeni, v časovni ga predstavljajo vrednosti v določenih trenutkih, v frekvenčni pa vrednosti pri določenih frekvencah. Filter deluje v frekvenčni domeni, vpliva na frekvence, ki jih želimo zavreči, ohraniti ali poudariti, torej spremeni amplitudo signala v določenih frekvencah. Zelo lepo lahko ponazorimo delovanje filtra z našimi usti. Izrecite samoglasnik, kot je recimo 'a', in nato z enakim tonom glasu izrecite še druge samoglasnike. S tem ne spreminjamo delovanja glasilk, le naš vokalni trakt. Glasilke ustvarijo signal s fiksnim frekvenčnim spektrom, vokalni trakt pa nato deluje kot filter, ki poudari nekatere dele spektra. Tu sem opisal le učinek filtrov na frekvenčno domeno signala, seveda filtri učinkujejo tudi na časovno domeno, a je ta učinek bolj kompliciran.

Osnovni tipi filtra so:

- **nizkoprepustni (Lowpass, LP)** prepusti frekvence do rezne frekvence f_c in oslabi frekvence nad f_c ;
- **visokoprepustni (Highpass, HP)** prepusti frekvence nad rezno frekvenco f_c in oslabi frekvence pod f_c ;
- **pasovnoprepustni (Bandpass, BP)** prepusti frekvenc med nižjo rezno frekvenco f_{cl} in višjo rezno frekvenco f_{ch} , frekvence nad f_{ch} in pod f_{cl} oslabi;
- **pasovnozavrnili (Bandreject, BR)** oslabi frekvence med nižjo rezno frekvenco f_{cl} in višjo rezno frekvenco f_{ch} , frekvence nad f_{ch} in pod f_{cl} prepusti;
- **zarezni (Notch)** oslabi frekvence v ozkem pasu okrog rezne frekvence f_c ;
- **resonančni (Resonant)** prepusti frekvence v ozkem pasu okoli rezne frekvence f_c ;
- **vseprepustni (Allpass)** prepusti vse frekvence, a spremeni fazo vhodnega signala.



Slika 2.1 Frekvenčni odzivi, vir: [7]

Druge vrste filtrov lahko opišemo kot kombinacijo teh osnovnih, recimo nizko prepustni filter z resonanco, glavnik (comb, več resonančnih) itd.

Sledi nekaj načinov uporabe raznih tipov filtrov: nizko prepustni z resonanco se uporablja v računalniški glasbi za simulacijo akustične resonančne strukture. Z visoko prepustnim odstranimo nezaželene nizke frekvence iz signala. S pasovno prepustnim lahko ustvarimo imitacijo telefonske linije. Zarezni je zelo uporaben za odstranjevanje nezaželenih frekvenc. Resonančni, kot pove že ime, se lahko uporabi za dodajanje resonance nekemu zvoku [7].

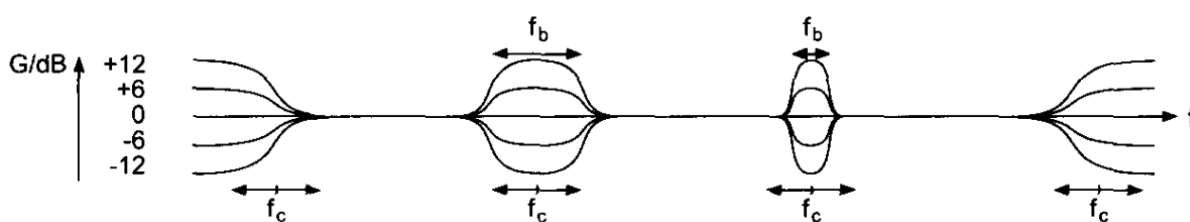
Glavne delitve med vrstami filtrov:

- **analogni in digitalni filtri:** v elektroniki so analogni filtri osnovni sestavni deli v napravah za procesiranje signalov. Digitalni filtri so sistemi, ki delujejo na vzorčenih diskretnih signalih;
- **časovno variantni in invariantni filtri:** diskreten sistem je časovno invarianten, če njegov izhod ni odvisen od časa oz. če se njegovo delovanje ne spreminja s časom;
- **linearni in nelinearni filtri:** diskretni sistem je linearen, če zadošča načelu superpozicije;
- **kavzalni in nekavzalni filtri:** sistem je kavzalen, če je izhod sistema odvisen le od trenutnega in preteklih vzorcev. Nekavzalen je, če je odvisen tudi od prihodnjih;
- **stabilni in nestabilni filtri:** sistem je stabilen, če je njegov izhod omejen za vsak omejen vhodni signal;

končen enotin odziv (KEO) in neskončen enotin odziv (NEO): enotin odziv NEO filtrov nikoli ne pade do nič, medtem ko enotin odziv KEO filtrov pade na nič v končnem času in traja $n + 1$ vzorcev, kjer je n red filtra. V praksi enotin impulz NEO filtrov pade pod neko mejo, pri kateri jih lahko zanemarimo, saj postane manjša od najmanjše predstavljljive amplitude. Sistemi, ki ustvarijo NEO ali KEO odziv, imajo pomembnejše razlike. NEO filtrom pravimo tudi rekurzivni filtri, saj za razliko od KEO filtrov za izračun naslednjega vzorca uporabijo tudi izhodne vzorce.

2.2 Izenačevalnik signala

Za razliko od tipov filtrov, ki sem jih opisal v prejšnjem podpoglavju, ki prepustijo ali oslabijo signal nad ali pod neko frekvenco, izenačevalniki oblikujejo frekvenčni spekter signala, tako da ojačajo ali oslabijo določene pase signala, medtem ko drugi ostanejo nespremenjeni. Poznamo dve osnovni obliki filtrov za izenačevanje. Prva je polični filter (shelving filter), ki pojača ali oslabi signal nad ali pod rezno frekvenco f_c . Druga je konični (peak filter), ki ojača ali oslabi frekvence v okolici rezne frekvence f_c . Širino pasu (bandwidth) določamo s parametrom f_b . Obstajajo tudi konični filtri s konstantnim Q -jem. Faktor Q je definiran kot razmerje med širino pasu in rezno frekvenco $Q = f_b/f_c$. Režno frekvenco spreminjamo, medtem ko ostaja Q faktor enak. To pomeni, da se širina pasu veča s tem, ko večamo rezno frekvenco, in obratno. Takšni filtri zvenijo bolj naravno, saj človeški sluh bolj natančno zaznava frekvenčne razlike pri nižjih tonih. Izenačevalniki so izredno pomembni pri mešanju več posnetkov skupaj, saj lahko z njimi iz posnetka odrežemo nepotrebne dele frekvenčnega spektra. Recimo pri mešanju govorne besede lahko porežemo nizke tone, saj tam govor vsebuje malo informacij [7].



Slika 2.2 Polični in konični filtri, vir. [7]

2.3 Dinamični razpon signala in njegov procesor

2.3.1 Dinamični razpon

Dinamični razpon je razmerje med najglasnejšim in najtišjim delom signala, merimo ga v decibelih. Najtišji deli so po navadi omejeni s šumom medija, v zvočnih sistemih je to elektroakustični šum naprav. V digitalnem svetu pa je to kvantizacijski šum, ki je odvisen od števila bitov signala. Dinamični razpon digitalnega zvoka je torej razmerje med kvantizacijskim šumom in najvišjo možno amplitudo signala. Dinamični razpon je pomemben v radijski in glasbeni produkciji. Posnetki z manjšim dinamičnim razponom zvenijo polnejše in glasnejše. Če mešamo več posnetkov skupaj, bodo nekateri detajli in tišji deli posnetkov prekriti z glasnejšimi. S procesorji dinamičnega razpona lahko to popravimo.

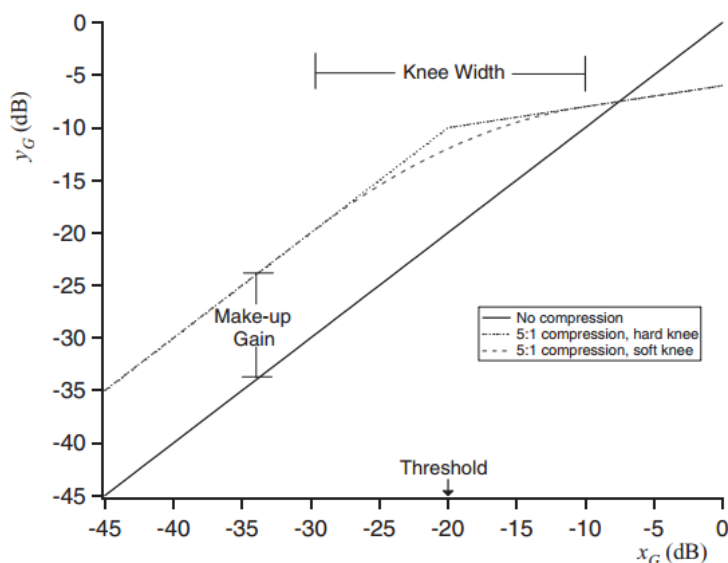
2.3.2 Procesor dinamičnega razpona

Procesor dinamičnega razpona signala spreminja dinamični razpon tako, da spremlja njegovo amplitudo in glede na nastavitve procesorja v nekem trenutku pojača ali oslabi signal. Najbolj pogosta oblika je kompresor dinamičnega razpona. Ko pride signal do neke določene meje, signal oslabi glede na določeno razmerje. Tišji deli ostanejo nespremenjeni, tisti, ki so čez mejo pa postanejo tišji. S tem se je zmanjšal dinamični razpon tega signala. V glasbeni produkciji ponavadi po kompresiji celoten signal še pojačamo, tako dobimo bolj prezenten zvok pri enaki jakosti, ki se zdi glasnejši.

Procesor dinamičnega razpona signala spreminja dinamični razpon tako, da spremlja njegovo amplitudo in glede na nastavitve procesorja v nekem trenutku pojača ali oslabi signal. Najbolj pogosta oblika je kompresor dinamičnega razpona. Ko pride signal do neke določene meje, signal oslabi glede na določeno razmerje. Tišji deli ostanejo nespremenjeni, tisti, ki so čez mejo, pa postanejo tišji. S tem se je zmanjšal dinamični razpon tega signala. V glasbeni produkciji po navadi po kompresiji celoten signal še pojačamo, tako dobimo bolj prezenten zvok pri enaki jakosti, ki se zdi glasnejši.

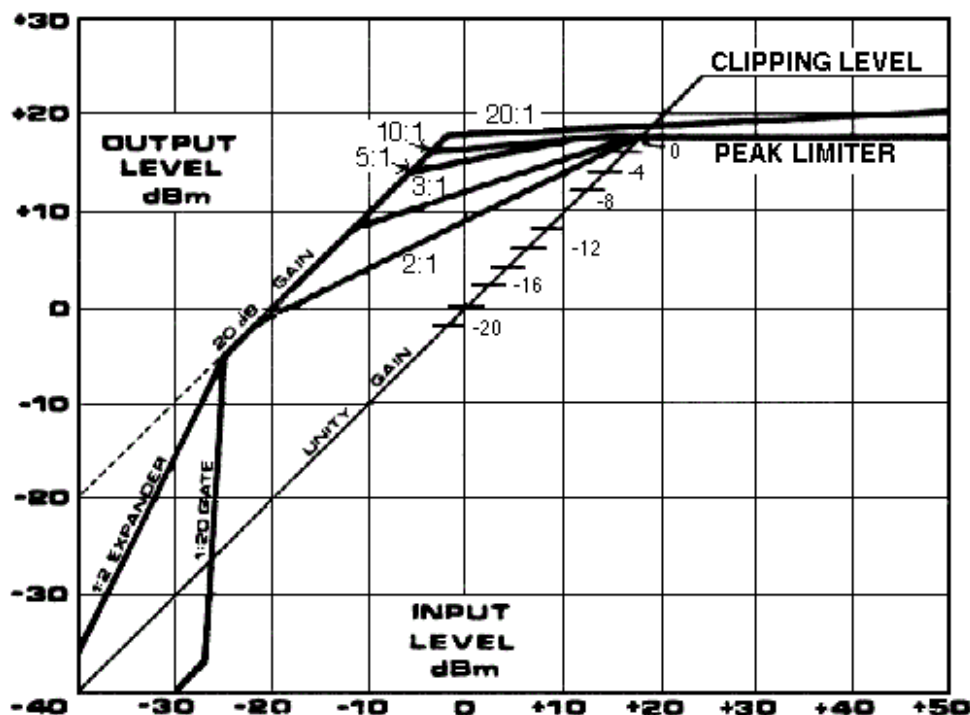
Oglejmo si natančneje delovanje kompresorja in najpogostejše parametre. Meja kompresije določi nivo jakosti signala, pri katerem se kompresija začne. Razmerje kompresije določa razmerje jakosti med izhodnim in vhodnim signalom, ki je presegel mejo. Na primer, če imamo razmerje 1 : 2 in je signal prekoračil mejo za 4 dB, bo izhodni signal oslavljen za 2 dB. Čas napada in sprostitve določata, kako hitro se kompresija začne, ko signal preseže mejo, in kako hitro preneha, ko signal pade pod mejo. Če bi uporabili takojšnje delovanje brez časovnega signala, bi popačili signal. Pred izhodom signala imamo možnost naknadnega pojačanja signala, da izenačimo izhodno jakost z vhodno. Širina kolena določa, ali uporabimo oster ali

zaokrožen prelom karakteristike kompresorja. Koleno je točka, določena z mejo kompresije, kjer se razmerje spremeni iz 1 : 1 v nastavljeno razmerje [2].



Slika 2.3 Karakteristika kompresorja, vir: [2]

Kompresija je le ena oblika procesorja dinamičnega razpona. Obstaja še ekspanzija, kompresija navzgor in ekspanzija navzdol. Ekspanzija se od kompresorja razlikuje le v tem, da uporabimo razmerje, večje od 1. Pri kompresiji so razmerja manjša od 1. Če pa obrnemo delovanje meje tako, da se procesiranje začne, ko signal pade pod njo, dobimo ekspanzijo navzdol, če je razmerje večje kot 1, in kompresijo navzgor, če je manjše od 1. Slednja je redko uporabna, saj dvigne prag šuma v posnetku. Če uporabimo zelo veliko razmerje, dobimo namesto kompresije limitiranje, torej signal sploh ne pride čez prag. Namesto ekspanzije navzdol pa dobimo efekt gate, jakost pade na nič, takoj ko pade pod mejo, to je uporabno za odstranjevanje šuma.



Slika 2.4 Razmerja procesiranja, vir: <http://www.sfu.ca/sonic-studio/handbook/Compression.html>

2.4 KEO filtri, linearna faza

Vrnimo se k filtrom. V diplomskem delu sem uporabil KEO filtre, zato bom le-te bolj natančno opisal. KEO filter, dolžine M , z vhodom $x(n)$ in izhodom $y(n)$, je definiran z enačbo (2.1).

$$\begin{aligned} y(n) &= b_0 X(n) + b_1 X(n-1) + \dots + b_{M-1} X(n-M+1) \\ &= \sum_{k=0}^{M-1} b_k x(n-k) = \left(\sum_{k=0}^{M-1} h(k) x(n-k) \right) \end{aligned} \quad (2.1)$$

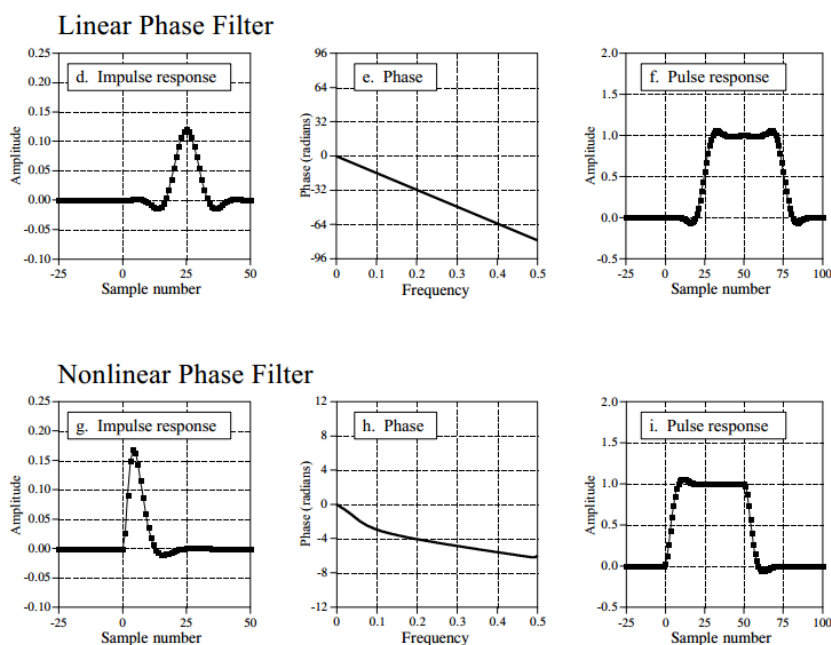
(b_k) so koeficienti filtra, $h(k)$ pa enotin odziv. Iz te enčabe vidimo končnost in kavzalnost filtra ter da je izhod filtra enak konvoluciji enotnega odziva z vhodom [1]. Filter lahko definiramo z njegovo sistemsko funkcijo (enačba (2.2)),

$$H(z) = \sum_{k=0}^{M-1} h(k) z^{-k} \quad (2.2)$$

ki je polinom stopnje $M-1$, spremenljivke z^{-1} . KEO filter ima linearno fazo, če za njegov enotin odziv velja enačba (2.3).

$$h(n) = \pm h(M-1-n) \quad n = 0, 1, 2, \dots, M-1 \quad (2.3)$$

Torej enotin odziv mora biti simetričen ali nasprotno simetričen [5].



Slika 2.5 Linearnost v povezavi z enotnim in faznim odzivom, vir: [6]

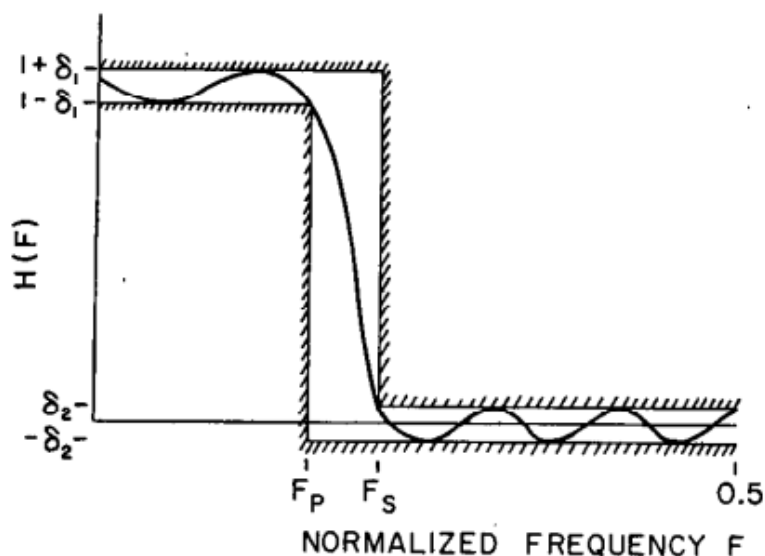
Poznamo štiri tipe filtrov z linearno fazo:

1. simetričen enotin odziv lihe dolžine,
2. simetričen enotin odziv sode dolžine,
3. nasprotno simetričen enotin odziv lihe dolžine,
4. nasprotno simetričen enotin odziv sode dolžine.

3. in 4. tip imata 90 stopinj faznega zamika, 2. tip ima ničlo pri frekvenci $F_s/2$, 3. tip ima ničlo pri frekvencah 0 in $F_s/2$, 4. tip pa ima ničlo pri frekvenci 0. 1. in 3. tip imata celoštevilski skupinski zamik.

2.5 Algoritem Parks-McClellan

Parks-McClellan je učinkovit iterativni algoritem za iskanje optimalnega Chebyshevega KEO filtra z linearno fazo. Cilj algoritma je minimizirati napako v prepustnem in neprepustnem pasu posebej, z uporabo Chebysheve aproksimacije [4]. Pri tej aproksimaciji razširimo iskano funkcijo s Chebyshevimi polinomi in nato razširitev odrežemo pri želenemu redu. Algoritem Parks-McClellan je variacija Remezovega algoritma, zasnovana za KEO filtre.



Slika 2.6 Prikaz parametrov algoritma Parks-McClellan, vir: [4]

Če iščemo nizko prepustni filter, algoritmu podamo F_P frekvenco konca prepustnega pasu, F_S frekvenco začetka neprepustnega pasu, δ_1 odmik v prepustnem pasu in δ_2 odmik v neprepustnem pasu. Z algoritmom lahko iščemo tudi bolj kompleksne filtre z več pasovi, vse pa podamo na podoben način.

2.6 Večpasovni procesor dinamičnega razpona

VPDR je orodje, ki procesira dinamičen razpon posameznih pasov frekvenčnega spektra. Lahko ga gledamo kot nadgradnjo 'navadnega' procesorja dinamičnega razpona in frekvenčnega izenačevalnika, saj rešuje nekatere probleme, ki jih imata. Če uporabimo izenačevalnik na nekem posnetku tako, da znižamo količino signala v nekem frekvenčnem pasu, kjer se pojavlja moteč presežek, bo to znižanje delovalo tudi, ko presežka ni. To pa lahko pokvari barvo zvoka v posnetku. Če uporabimo procesor dinamičnega razpona, tako da skompresiramo nek posnetek, ki vsebuje širok razpon frekvenc, a jih je preveč le na enem pasu, bo procesor potisnil dol tudi tiste frekvence, ki jih ni preveč. Če ima posnetek šum in uporabimo za njegovo

odstranitev procesor dinamičnega razpona kot gate efekt, bo to zelo slišno in moteče. Z izenačevalnikom pa tudi težko odstranimo šum, saj ga lahko uporabimo le tako, da ne odstranimo delov frekvenčnega spektra, ki so za nas pomembni. Izenačevalnik je frekvenčno selektiven, a ni dinamično selektiven, medtem ko je procesor dinamičnega razpona dinamično selektiven, a ni frekvenčno selektiven. VPDR rešuje ta dva problema tako, da signal najprej s pasovno prepustnimi filtri (razen prvi in zadnji, ki sta nizko in visoko prepustna) razbije na nekaj pasov in vsak pas procesira z VPDR posebej. Tako lahko dinamični razpon posameznih frekvenčnih pasov procesiramo neodvisno in z različnimi parametri. Glavni parametri VPDR so vsi parametri, ki sem jih naštel v poglavju o procesorjih dinamičnega razpona, za vsak pas posebej. Ima še parametre za prilagajanje pasov ter za hkratno nastavljanje parametrov procesorjev.

2.7 Matlab

Matlab je okolje za numerično računanje in programski jezik četrte generacije. V diplomskem delu sem ga uporabil za zasnovo pasovno prepustnih filtrov. Najpomembnejše funkcije, ki sem jih uporabil, so: `firpm` in `firpmord`. `Firpm` je funkcija za izračun enotnega odziva optimalnega filtra z linearno fazo po algoritmu Parks-McClellan. Podamo ji prepustne ter zaporne predele, deviacijo (odstopanje od želene frekvenčne karakteristike) v posameznih predelih, red filtra ter vzorčno frekvenco. Vrne nam enotin odziv filtra z minimalno razliko od zelenega. Ker je za optimalnost filtra potreben tudi dovolj velik red, uporabimo funkcijo `firpmord`, podamo ji želene karakteristike in vrne nam približni red za optimalen filter. Uporabil sem tudi `fvtool`, orodje za vizualizacijo filtrov, ki za podan enotin odziv filtra izriše njegov frekvenčni in fazni odziv, fazni in skupinski zamik, enotin ter stopnični odziv ter še nekaj drugih uporabnih informacij. Za izračun skupnega frekvenčnega odziva vseh filtrov sem uporabil funkciji `fft` ter `ifft`. Enotin odziv vsakega filtra sem s `fft` pretvoril v frekvenčno domeno, jih seštel in na koncu z `ifft` pretvoril spet v časovno domeno. S tem sem dobil enotin odziv seštevka vseh filtrov, z uporabo `fvtool` sem lahko opazoval njegov frekvenčni in fazni odziv. Enotine odzive zasnovanih filtrov sem z `audiowrite` shranil kot `.wav` datoteke za enostavno prenašanje v okolje SuperCollider 3.6.

2.8 SuperCollider 3.6

SuperCollider je programski jezik in okolje za realnočasovno sintezo zvoka ter algoritmično kompozicijo. Vsebuje tudi ogrodje za DPS, ki sem ga s pridom uporabil za to diplomsko delo.

Najpomembnejši objekti v SC, ki sem jih uporabil, so: Convolution2, Compander in mnogo objektov za grafični vmesnik. Convolution2 je objekt za FFT konvolucijo signala in jedra. Podamo mu vhodni signal, buffer z jedrom filtra ter velikost FFT okvirja. Compander je procesor dinamičnega razpona, ki zmore kompresijo, ekspanzijo, ekspanzijo navzdol in kompresijo navzgor. Če uporabimo dovolj veliko razmerje, ga lahko uporabimo tudi za limitiranje in kot efekt gate. Uporabil sem tudi veliko različnih objektov, da sem ustvaril grafični vmesnik. Uporabil sem userView za izrisovanje funkcij, ki nastavljajo parametre Companderjev, in veliko število EZSlider ter EZKnob objektov za kontrolo nad parametri funkcij. Zelo priročna lastnost SuperColliderja je večkanalna razširitev. Ta deluje tako, da če objektu podamo tabelo za nek parameter, se bo avtomatsko ustvarilo več kanalov tega objekta. Zelo pomemben objekt v SC je SynthDef, s katerim definiramo Synthe, ti so enote za procesiranje ali sintetiziranje signala.

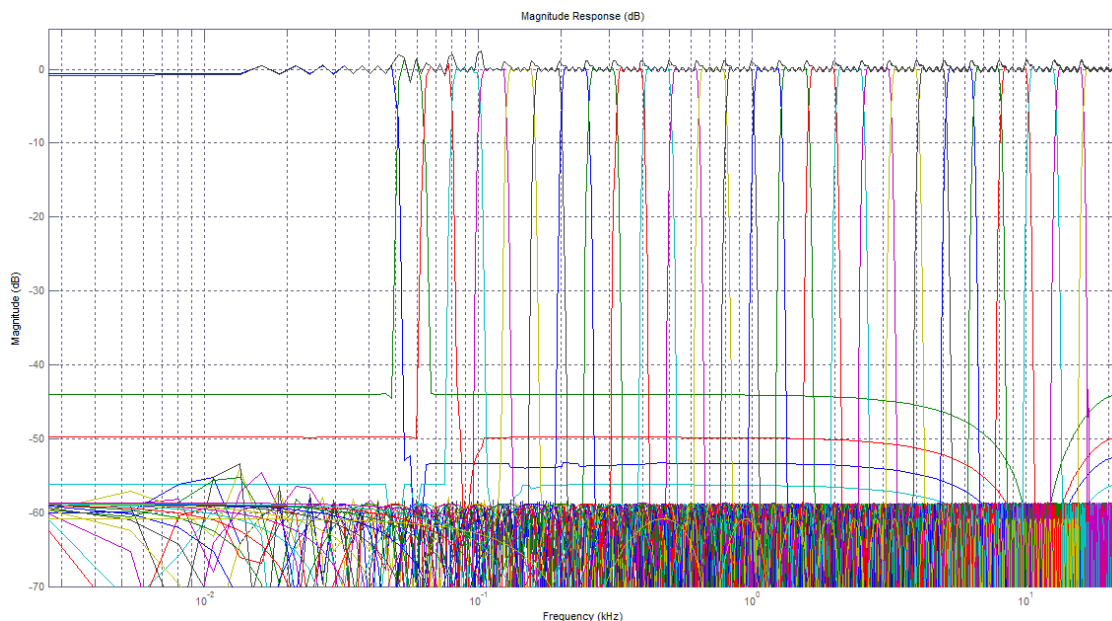
Poglavje 3 Izvedba

3.1 Cilj diplomskega dela

Moj cilj je ustvariti nadgradnjo večpasovnega procesorja dinamičnega razpona signala. Večpasovni procesorji običajno frekvenčni spekter razdelijo na 3 do 5 pasov, na vsakem pasu je procesor dinamičnega razpona signala z neodvisnimi nastavitvami. Cilj je, da bi lahko čim bolj zvezno nastavljal zgornjo in spodnjo mejo kompresije ter ekspanzije ter spodnje in zgornje razmerje preko celega frekvenčnega spektra. Odločil sem se, da razdelim spekter na 30 tretjino oktave širokih pasov. Njihove parametre in pojačanje/oslabitev pa nastavljam z dvema arcus tangens funkcijama, ki bosta delovali kot visoki in nizki polični filter, ter eno gaussovo funkcijo, ki bo delovala kot konični filter za srednje frekvence.

3.2 Zasnova filtrov

Na začetku izdelave diplomskega dela sem nameraval celotni dinamični izenačevalnik implementirati v SuperColliderju, a kmalu sem ugotovil, da njegovi filtri niso uporabni za dinamični izenačevalnik. Imajo prepočasen padec od prepustnega do neprepustnega pasu in njihov neprepustni pas ima premajhno oslabitev. Ti dve lastnosti povzročata dva problema. Če pojačamo enega od filtrov, je slišna pojačitev celotnega spektra. Na procesor dinamičnega razpona vpliva preveč signala, kar izniči cilj natančnosti in uporabo tretjino oktave širokih filtrov za dinamični izenačevalnik. Zato sem se določil, da moram zasnovati boljše filtre. Uporabil sem KEO filtre. NEO filtri imajo prednost, da za isti frekvenčni odziv potrebujejo veliko manjši red kot KEO filtri, vendar je njihova zasnova veliko kompleksnejša kot zasnova KEO filtrov. Po branju člankov, v katerih opisujejo snovanje NEO pasovno prepustnih filtrov, širokih tretjino oktave, sem izvedel, da taki filtri v eni implementaciji niso stabilni pri nizkih frekvencah in nimajo dovolj oslabitve nad prepustnim pasom [1], v drugi pa, da imajo prepočasen padec v prehodnem pasu [3]. Razlog za zavrnitev NEO filtrov je bil tudi ta, da SuperCollider nima objekta za realizacijo NEO filtrov. Torej sem sprejel kompromis, filtri ne bodo realnočasovni, ampak bo njihovo snovanje veliko lažje ter natančno. Še ena prednost KEO filtrov je, da imajo linearno fazo, čeprav ta ni nujna v glasbeni produkciji, a je pri NEO filtrih popolna linearnost faze nemogoča.

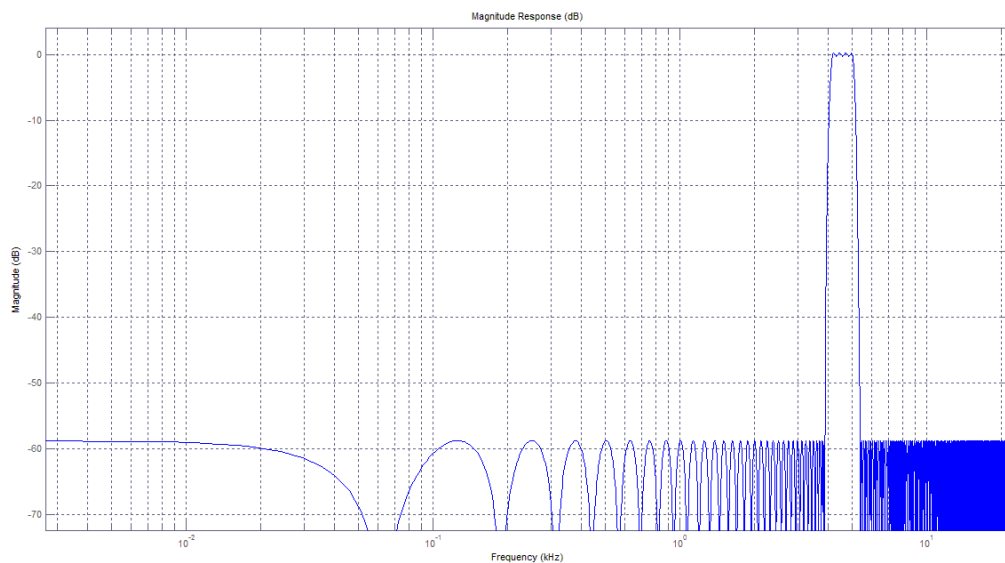


Slika 3.1 Frekvenčni odzivi 27 filtrov

Odmik v prepustnem pasu sem nastavlil na 0,5 dB, oslavljenje v neprepustnem pasu na – 60,0 dB. Točke filtra sem določil tako:

```
passPrva=20*e^(i-1+stis);
passDruga=20*e^(i-stis);
stopPrva=20*e^(i-1+stis-pas);
stopDruga=20*e^(i-stis+pas);
f=[stopPrva passPrva passDruga stopDruga];
```

$e = \sqrt[3]{2}$, $stis$ sem podal stisk širine filtra, pas pa je širina prehodnega pasu. Z veliko poskušanja sem pri vrednostih $stis = 0,09$ in $pas = 0,3$ dobil dobro seštevanje filtrov.

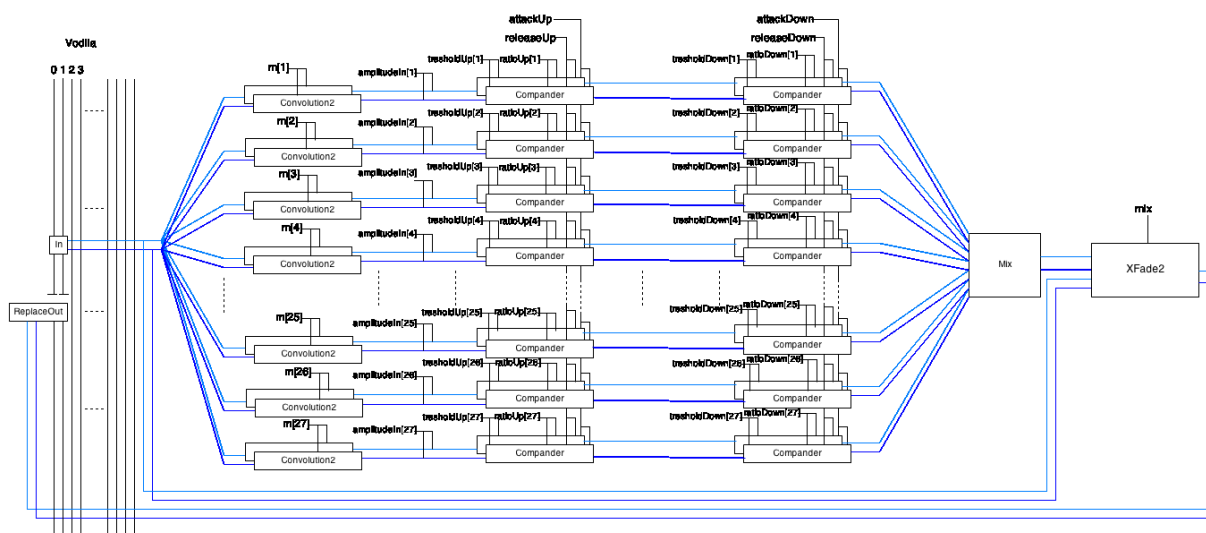


Slika 3.2 Frekvenčni odziv enega filtra

Redi filtrov so sodi in omejeni na $16.382 = 2^{14} - 2$, dolžina filtra mora biti zaradi FFT konvolucije potenca 2. Tako dolgi filtri so porabili približno 15 minut za izračun. Tako sem dobil 1. tip filtrov z linearno fazo, saj algoritem Parks-McClellan vrne simetričen enotin odziv, njegova dolžina pa je liha zaradi sodega reda. Filtri pod 100 Hz niso več optimalni pri redu 16.382, a so do 50 Hz še sprejemljivi. Zato sem prvi filter, ki ni pasovnoprepustni, ampak nizkoprepustni, naredil kar do 50 Hz. S tem sem dobil 27 filtrov in ne 30, kot načrtovano. Ker je fazni zamik filtra vedno $\frac{n-1}{2}$, kjer je n dolžina enotinega odziva, in ker so enotini odzivi višjih filtrov krajši od nižjih, ima vsak filter drugačen zamik. Zato sem jih vse zamaknil in dopolnil z ničlami tako, da so njihovi sredinski koeficienti poravnani. S tem imajo vsi enak fazni zamik in ko so seštet skupaj, ima seštevek linearno fazo. Vsakega od filtrov sem s fft pretvoril v frekvenčno domeno, jih tam seštel in pretvoril nazaj v časovno, da sem lahko seštevek izrisal skupaj s filtri in tako primerjal, kako se seštevajo z različnimi parametri.

3.3 Implementacija dinamičnega izenačevalnika

Implementacijo sem napisal v programskem jeziku SuperCollider 3.6 s pomočjo istoimenskega programskega okolja. Za hranjenje koeficientov sem uporabil objekt Buffer. Številke Bufferjev sem shranil v tabelo, zato da jo lahko uporabim za večkanalno ekspanzijo. Nato sem definiral glavni SynthDef za dinamični izenačevalnik.

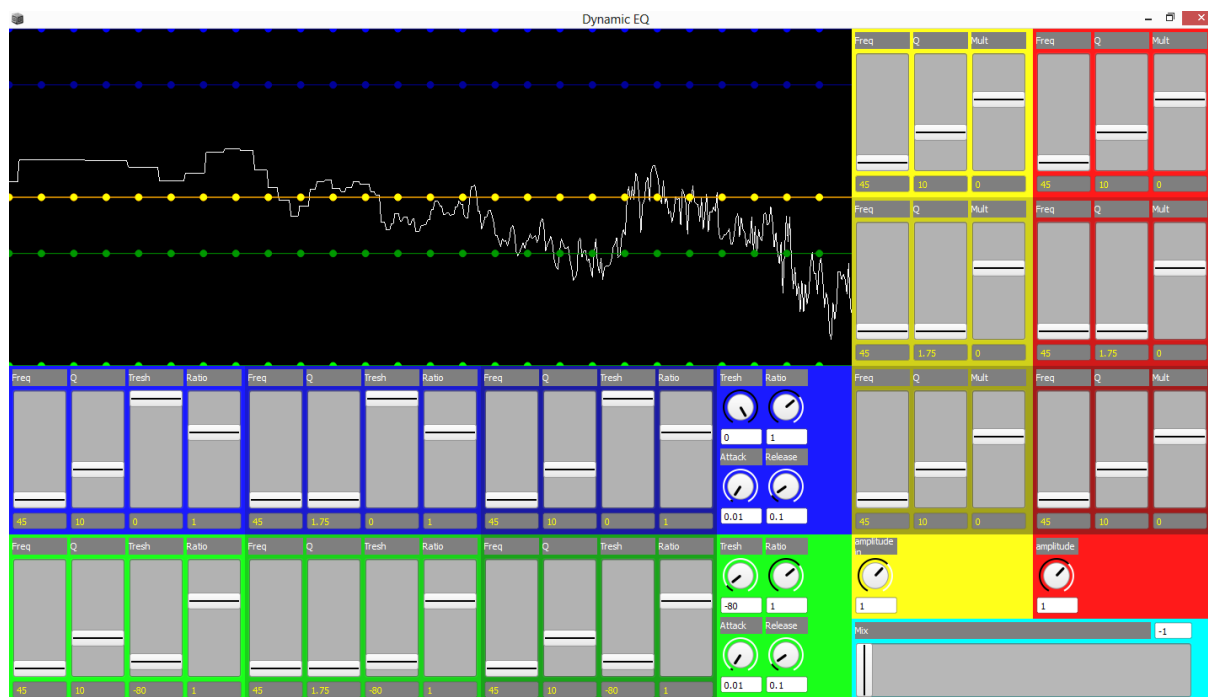


Slika 3.3 Shema implementacije

Z objektom In sem definiral, da je vhod v Synth iz vodil 0 in 1. Z ukazom .names objekta Control sem ustvaril tabele parametrov za pasove, ki jih lahko nastavljamo izven Syntha. Tabele parametrov so: vhodna in izhodna pojačitev, zgornja in spodnja meja procesiranja ter zgornje

in spodnje razmerje. Nato sem objektu Convolution2 podal stereo vhod kot tabelo in tabelo števil Bufferjev. Zaradi večpasovne ekspanzije se s tem ustvari tabela 26 tabel dveh kanalov. Vsaka dvojica kanalov je konvolirana z Bufferjem, napolnjenim s koeficienti filtra. Potem sem pomnožil tabelo 26 tabel signalov s tabelo vhodnih pojačitev. Zatem sem definiral prvi Compander, ta bo skrbel za zgornjo kompresijo in ekspanzijo. Podal sem mu novo tabelo 26 tabel za vhodni in kontrolni signal, tabelo zgornjih mej, tabelo razmerij procesiranja ter čas napada in sprostitve, ki ju lahko ravno tako nastavljam izven Synthaa. Nato sem definiral še drugi Compander, ki bo skrbel za spodnjo kompresijo in ekspanzijo oziroma kompresijo navzgor in ekspanzijo navzdol. Podal sem mu izhod prejšnjega Companderja ter enake tabele in parametre kot prejšnjemu. Nato sem uporabil objekt XFade2 tako, da lahko s parametrom mix kontroliram, koliko procesiranega in koliko čistega signala gre na izhod. Če je $\text{mix} = -1,0$, dobim na izhod le procesiran signal, če je $\text{mix} = 1,0$, dobim le čist signal, če pa je vmes med vrednostma, se signala zmešata. Programski jezik sam poskrbi, da sta oba signala enako zamaknjena. Na koncu definicije Synthaa sem z objektom ReplaceOut nastavil, da gre izhod Synthaa na vodili 0 in 1. ReplaceOut prepiše vodilo, da je na njem le izhod dinamičnega izenačevalnika. Definiral sem še dva testna Synthaa, enega, ki predvaja posnetek z objektom PlayBuf, ter enega, ki ustvari šum.

3.4 Grafični vmesnik



Slika 3.4 Grafični vmesnik

Grafični vmesnik je sestavljen iz 6 podoken. Prvo je prikazovalnik frekvenčnega spektra ter izrisovalnik krivulj. Za prikaz frekvenčnega spektra sem uporabil objekt `FreqScopeView`. Krivulje nastavljamo s preostalimi podokni, modro in zeleno podokno sta za nastavljanje meje in razmerja procesiranja, rumeno in rdeče podokno sta za nastavljanje vhodnega in izhodnega pojačanja. Svetlomodro podokno je za nastavljanje mešanja med procesiranim in čistim signalom. Naslednja enačba je uporabljena za izračun pojačitve i-tega pasa.

Sestavljena je iz dveh arcustangens in ene gaussove funkcije, tako da spominja na tipično obliko

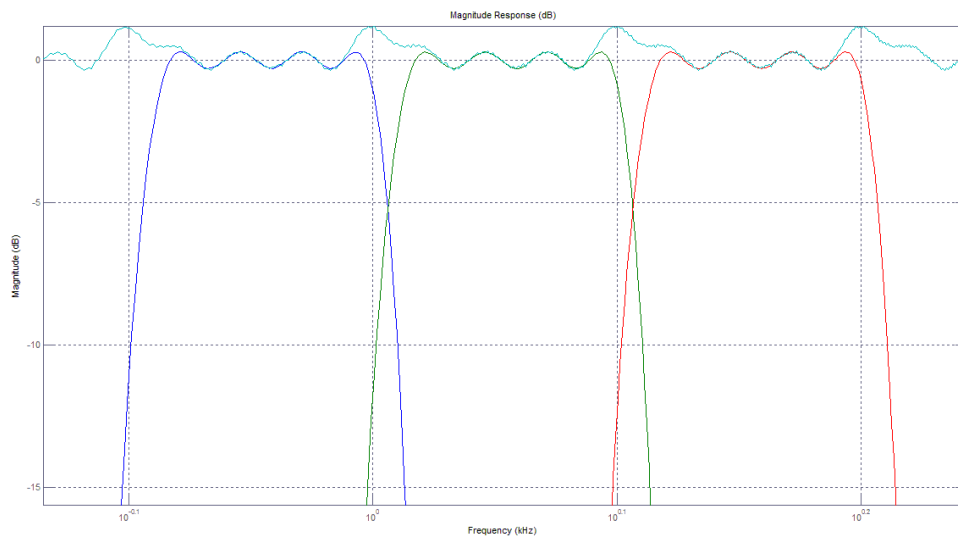
$$\begin{aligned}
 & \text{amplitude}(i) \\
 &= \left(\frac{\tan^{-1}(-\text{ampCtlShelfLowQ} * (i - \text{ampCtlShelfLowFreq}))}{\pi} \right. \\
 & \quad \left. + 0.5 \right) (\text{ampCtlShelfLowMult} - 1) \\
 & \quad + (\text{ampCtlPeakMult} - 1) e^{-\frac{(i - \text{ampCtlPeakFreq})^2}{2\text{ampCtlPeakQ}^2}} \\
 & \quad + \left(\frac{\tan^{-1}(\text{ampCtlShelfHighQ} * (i - \text{ampCtlShelfHighFreq}))}{\pi} \right. \\
 & \quad \left. + 0.5 \right) (\text{ampCtlShelfHighMult} - 1)
 \end{aligned} \tag{3.1}$$

izenačevalnika, z dvema poličnima filtroma in enim koničnim. Vse druge krivulje so izračunane na enak način, le da uporabljajo druge parametre.

Del kode za grafični vmesnik je veliko daljši kot drugi deli, saj je bilo treba definirati veliko objektov in parametrov. Definirati je treba tudi risalno funkcijo, ki izrisuje krivulje preko frekvenčnega spektra. Največ vrstic kode pa so zasedle definicije 53 drsnikov in gumbov. Ko jih definiramo, jim moramo podati podokno, v katerega naj se izrišejo, velikost, akcijo, ki se izvede pri premiku drsnika ali gumba, ter `ControlSpec`. S `ControlSpec` določimo razpona števil, ki jih lahko nastavljamo, minimalni korak, razporejenost vrednosti (linearna ali eksponentna) in začetno vrednost. V akciji vsakega drsnika in gumba sem zapisal vrednost drsnika v parameter, posodobil tabelo vrednosti in posodobil parametre `Syntha`.

Poglavje 4 Evaluacija dela

Dinamični izenačevalnik deluje, kakor je bilo načrtovano. Filtri, ki so bili zasnovani, imajo zelo ozke prehodne pase, malo odmika v prepustnem pasu ter dobro oslabijo v prepustnem.



Slika 4.1 Prekrivanje filtrov

Filtri se med sabo dobro seštevajo, saj je odmik seštevka na delu, kjer so filtri optimalni, le 1,2 dB. Prvi štirje filtri niso optimalni, odmik seštevka na tem delu je največ 2,6 dB.



Slika 4.2 Prekrivanje neoptimalnih filtrov

Za doseg boljšega rezultata bi bilo treba definicije filtrov zastaviti kot optimizacijski problem, saj je odmik seštevka filtrov odvisen od nekaj parametrov. Program se izkaže zelo dobro pri odstranjevanju šuma s posnetka. To naredimo tako, da dvignemo spodnjo mejo na celotnem frekvenčnem spektru toliko, da je nad jakostjo šuma. Spodnje razmerje nastavimo na 10 na celotnem spektru. Pri tej nastavitvi vsak od procesorjev deluje kot gate efekt, ko pade signal pod neko jakost, ga procesor še dodatno zniža. Z malo igranja z nastavitvami posameznih pasov in nastavitvami časa napada in sprostitve dobimo dobre rezultate. Program deluje bolje kot navaden gate efekt. Velik problem programa pa je njegov grafični vmesnik, saj je težko natančno nastavljati nastavitve posameznih pasov z okornimi funkcijami.

Uporabo programa sem demonstriral sodelavcem na Radiu Študent. Bili so zadovoljni z delovanjem, a jih je zelo motila okornost in nenatančnost grafičnega vmesnika. Predlagali so mi grafični vmesnik, v katerem bi lahko z miško premikal vsak del funkcije posebej in bi se nastavitve izpisale, ko bi bila miška na nekem pasu za boljšo preglednost.

Poglavje 5 Zaključek

Za diplomsko delo sem zasnoval dinamični izenačevalnik. Ta razbije frekvenčni spekter signala na 27 pasov in nato procesira dinamični razpon vsakega posebej ter jih na koncu združi. V programskem okolju Matlab sem z algoritmom Parks-McClellan zasnoval 25 tretjino oktave širokih pasovno prepustnih pasov, enega nizko prepustnega do 50 Hz in enega visoko prepustnega od 16.255 Hz naprej. Filtre sem prenesel v programsko okolje SuperCollider 3.6 in tam sprogramiral razbitje signala na 27 signalov, vsakega se konvolira s filtrom ter nato procesira procesorjem dinamičnega razpona. Nastavitve procesorjev nastavljam z grafičnim vmesnikom, s katerim upravljamo 6 funkcij, ki spominjajo na navaden izenačevalnik. Imenuje se dinamični izenačevalnik, ker z njim procesiramo dinamični razpon posameznih delov frekvenčnega spektra, podobno kot z navadnim izenačevalnikom spreminjamo jakost posameznih delov spektra.

Uporabnost dinamičnega izenačevalnika je zelo široka. Lahko se ga uporablja za odstranjevanje šuma ali zato, da ekspanziramo nekatere dele zvočnega materiala, kot recimo udarce činel v glasbi, da pridejo bolj v ospredje. Lahko zelo natančno kompresiramo posamezne dele frekvenčnega spektra in dobimo bolj transparenten efekt, kot če bi enako kompresirali z navadnim kompresorjem. Če kompresiramo nizke in visoke tone, ko se celoten signal zniža, dobimo bolj enakomerno tišanje preko frekvenčnega spektra, saj nizke in visoke tone dojemamo tišje kot srednje pri manjši jakosti.

Diplomsko delo bi lahko nadgradil s tem, da bi izenačevalnik naredil realno časoven, a za to bi moral kompromizirati pri natančnosti. Pri raziskovanju nisem zasledil filtrov, ki bi v realnem času bili enako strmi in z dovolj dobrim oslavljenjem. Dobro bi bilo tudi implementirati kot VST vtičnik, da bi bil uporaben v okoljih za delo z digitalnim zvokom. Še ena možna nadgradnja je nadgradnja grafičnega vmesnika. Treba bi bilo najti način, kako narediti grafični vmesnik priročen, nazoren, enostaven in natančen. Izenačevalnik bi bil še bolj uporaben, če bi imel možnost, da bi lahko med delom zaprl večino pasov, tako da bi slišal le del spektra, katerega nastavitve trenutno nastavljam.

Literatura

- [1] Antonina Geras, Tomasz Starecki, *Properties of digital 1/3-octave filters implemented according to ANSI S1.11*, Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments 2012
- [2] Dimitrios Giannoulis, Michael Massberg, Joshua D. Reiss, *Digital dynamic range compressor design: A tutorial and analysis*, J. Audio Eng. Soc., zv. 60, št. 6, 2012 June
- [3] Janusz M. Mosakowski; Andrzej Podgórski, *Implementation of 1/1 octave and 1/3 octave filters in digital signal processor*, XVII IMEKO World Congress, June 22–27, 2003, Dubrovnik, Croatia
- [4] Thomas W. Parks, James H. McClellan, *Chebyshev approximation for nonrecursive digital filters with linear phase*, IEEE Transaction on circuit theory, zv. 19, št. 2, Marec 1972
- [5] John G. Proakis, Dimitris G. Manolakis, *Digital signal processing: principles algorithms and applications*, New Jersey, Prentice-Hall, 1996
- [6] Steven W. Smith, *The scientist and engineer's guide to digital signal processing*, San Diego, California Technical Publishing, 1997-1999
- [7] Udo Zolzer, *DAFX – Digital audio effects*, England John Wiley & Sons 2002

